Machine Learning and RSF

A Quick Overview of Key Technical Elements in DCYF's Implementation of Rapid Safety Feedback

Table of Contents

Introduction 1
Machine Learning 2
Supervised Learning 2
Unsupervised Learning 2
Tree-Based Modeling Methods: Decision Trees and Random Forests 3
NH RSF Model Implementation 4
Model Development and Evaluation 5
The Datasets Used in the Model's Construction and Evaluation 5
Choosing a Modeling Technique and Trial Testing 5
Applying New Data to the Model for Scoring 5
Model Retraining and Tuning 5
Appendix 6
Model Interpretability vs. Complexity: A Tradeoff 6
Overfit and Underfit Models: The Bias-Variance Tradeoff 6
Bonferroni's Principle 6
Works Cited 7

Introduction

The following is intended to be an introduction to the machine learning concepts and methods used in the construction of the Rapid Safety Feedback (RSF) random forest predictive model application in use at NH DCYF. The aim of this document is not to be an all-encompassing technical text, but rather, to provide a gentle introduction to essential topics. This document will look to familiarize readers with key machine learning terminology and concepts, explain how the model in use functions, and provide an overview of the process used in the model's construction, evaluation, and scoring.

Machine Learning

As Lantz defines it, machine learning is "The field of study interested in the development of computer algorithms to transform data into intelligent action...".[1] It is a wide-reaching field which has developed with the growth in available recorded data. The field encompasses elements from statistics, artificial intelligence, and computer science. Effectively, the process of machine learning revolves around discerning patterns or relationships among input data (training a model) and abstracting from it to output some form of inductive inference, which may lead to insight. This pattern and relationship discovery process relies on using advanced algorithms to fragmentize the input data. Machine learning is often used on tasks which are considered too complex to program outright or tasks which may be beyond the cognitive capabilities of human beings to conceptualize in terms of intricacies.[2] Machine learning methods can be broken down into two main categories of learning: supervised learning and unsupervised learning. Each subgroup has its own domain of application.

Supervised Learning

In supervised learning, the data used in training the model has a specific dependent target variable or series of labels associated with it, where the aim is predicting this variable in a new dataset.[3] This target may be categorical or quantitative. As Shalev-Shwartz and Ben-David put it, "In such cases, we can think of the [supervised learning] environment as a teacher that 'supervises' the learner by providing the extra information (labels).".[4]

A simple example of supervised learning may be an instance where the problem a model is trying to solve may be classifying emails into the categories of junk/spam and not junk/spam. In order to achieve this, the model is trained on a dataset of emails where each email has a corresponding target field containing its classification group (is it spam or not?). Once the model is trained, a new dataset of unclassified emails can be run through the model for scoring, which will assign a class prediction for each email in the new dataset based on what was abstracted from the training dataset.

In regard to the RSF project, the random forest model in use can be described as a supervised machine learning technique.

Unsupervised Learning

In unsupervised learning, there may be no dependent target variable associated with the data and the goal of this type of learning is rooted in gaining descriptive synopses and insights from a dataset. Often times, such methods are used to understand which subsets of a larger dataset can be clustered together based on similarities and patterns in the variables.

An example of unsupervised learning may be a case where a company wishes to look at customer and sales data and try to cluster together customers based on their purchasing habits.

---

[1] (Lantz 2015, 3)
[2] (Shalev-Shwartz and Ben-David 2014, 22)
[3] (James, et al. 2013, 26)
[4] (Shalev-Shwartz and Ben-David 2014, 23)

Tree-Based Modeling Methods: Decision Trees and Random Forests

In order to understand the specifics of the model used in NH's RSF implementation, it's important to understand some basic elements of decision trees and random forests, which are types of supervised learning models.

Decision tree models can be applied to classification and regression based predictive problems. In effect, as Lantz explains, decision trees work by splitting "...the data into subsets, which are then split repeatedly into even smaller subsets, and so on and so forth until the process stops when the algorithm determines the data within the subsets are sufficiently homogenous, or another stopping criterion has been met.".[5] The trees start with a root node and the data is split until it reaches a leaf containing the target variable. Along the way from the root node to the leaf, the tree is partitioned into branches based on the dataset's variables. Ideally, the branch split order is based on those variables with the highest predictive impact for each branch node.

An example of a simple decision tree can be seen below.[6] In it, the tree is used to determine whether a papaya will be tasty or not.

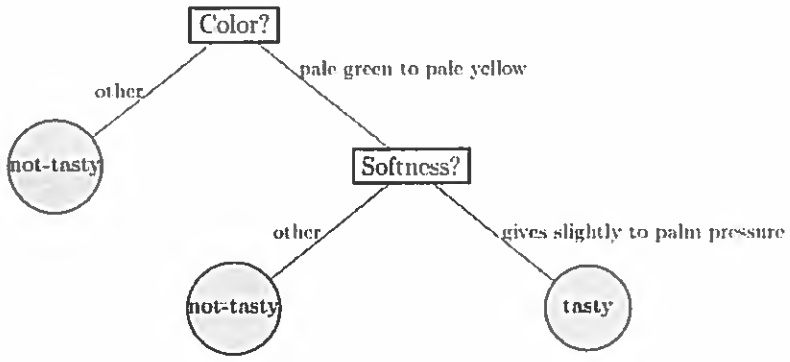

Figure 1: Decision Tree from Shalev-Shwartz and Ben-David 2014, pg. 250

Models built with decision trees are often easy to visualize and interpret. Despite this, their predictive power tends to suffer if the amount of variables that make up the training dataset is immense or if the dataset itself contains noise, skewness, or significant variance.[7] As such, the results of decision trees may vary highly based on what the training dataset is comprised of. For example, if a training dataset was split into two equal parts at random and two decision trees were constructed, each using one of the dataset partitions respectively, it is very possible that the two decision trees might end up looking quite differently compared to each other due to the specific variations captured in each training dataset partition.

---

[5] (Lantz 2015, 127)
[6] (Shalev-Shwartz and Ben-David 2014, 250)
[7] (Williams 2011, 205)

The predictive power of tree methods can be improved significantly by introducing randomness into the model building process with a statistical method called bootstrapping and by running more tree iterations in a method called ensembling. Essentially, bootstrapping implies working with random samples of the training dataset instead of using it as a whole. Both bootstrapping and ensembling techniques are applied when building random forest models.

Random forest models are ensembles of decision trees (often hundreds or thousands), where each decision tree is comprised of a random sample of data records from the complete training dataset and each branch split on every decision tree is chosen from a small random sample of all the available data variables.[8] After each tree is run, the most common prediction from all the decision trees is selected as the final model prediction. Effectively, each decision tree gets a vote in choosing the final prediction.

An example visual of a random forest model can be seen below.[9]

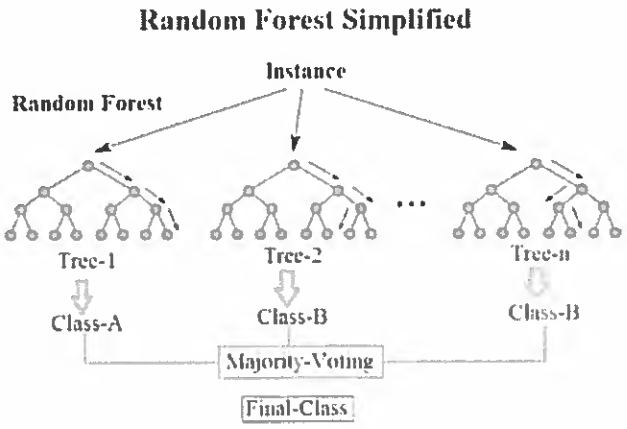

Figure 2: Random Forest Visualization from Reinstein n.d., KDnuggets

In all, due to bootstrapping and ensembling, random forest models tend to be more robust to the effects of noise, skewness, and variance as compared to a single decision tree.[10] As a tradeoff however, due to the complexity linked with constructing a random forest, much of the interpretability associated with using a single decision tree in terms of insight on how each node split was made is lost.

NH RSF Model Implementation

As has previously been illustrated, the RSF model can be described as a random forest model, which is a type of supervised machine learning model. What follows will be an overview of what is known about the specific model built for RSF, based on communications with Eckerd and Mindshare.

[8] (Williams 2011, 248)
[9] (Reinstein n.d.)
[10] (Williams 2011, 246)

Model Development and Evaluation

The Datasets Used in the Model's Construction and Evaluation

The RSF model has been built based on a dataset containing past historical data contained in NH's SACWIS and supplementary data on youth with serious injuries or fatalities. This dataset was partitioned into two subset datasets:

1. A training dataset (75% of the total dataset)
2. A testing dataset (25% of the total dataset)

The training subset was used in constructing the model, while the testing subset was used in evaluating the model's predictive accuracy.

Choosing a Modeling Technique and Trial Testing

In all, it has been communicated that three different modeling techniques where tried before one was selected for use in the current model:

1. A Decision Tree Model
2. A Random Forest Model
3. A Support Vector Machine Model[11]

Each trial iteration was evaluated using the holdout testing data and the random forest technique was selected as it provided the most accurate predictive power.

Applying New Data to the Model for Scoring

The completed model is applied for risk scoring to a scoring dataset on a daily basis. This scoring dataset is comprised of youth who meet the qualifications as laid out in the problem statement listed on the RSF Portal. At the time of this writing, these are youth with a current accepted referral who are "...known to the Department from a prior accepted report, regardless of finding, within 12 months of that previous accepted report.".[12]

Once run through the model, the youth comprising this scoring dataset are given a predicted risk score. This risk score is determined on a scale from 0 to 100, where scores of 50-100 are classified as "High Risk" and are listed on the Prediction section of the RSF portal. It is possible for a youth's risk score to vary each day it is present in the scoring dataset, depending on any updates made in the data extract from NH's SACWIS.

Model Retraining and Tuning

It has been communicated from Eckerd and Mindshare that the model will be retrained on a quarterly basis to evaluate predictive accuracy. Additional tweaks and tuning may be made with feedback from NH DCYF.

---

[11] Information on this technique will not be covered, as its understanding is outside the scope of this document.
[12] The full problem statement can be found on the RSF portal's homepage.

Appendix

What follows are a couple more useful concepts to understand about machine learning, however, they may be considered ancillary to what has been covered above.

Model Interpretability vs. Complexity: A Tradeoff

There are a myriad of modeling techniques which could be chosen to work on a given problem, each with its own strengths and weaknesses which need to be weighed based on the unique needs of the problem scope. Generally, in regard to modeling, there are tradeoffs between a model's interpretability and complexity. As a modeling technique becomes more complex in nature, its ability to be humanly interpreted diminishes. On the flip side of the coin, generally, as the model's complexity increases to a point, its accuracy also increases.[13]

Overfit and Underfit Models: The Bias-Variance Tradeoff

In many instances, models may suffer from being overfit or underfit. As Hawkins puts it, "Overfitting is the use of models or procedures that violate parsimony — that is, that include more terms than are necessary or use more complicated approaches than are necessary.".[14] In general, overfit models pick up too much noise or randomness that may be inherent in the training dataset and tend to over-abstract learned patterns as a result. These over-abstracted patterns do not generalize well to new data, causing the model's predictive power to suffer. This is due to what's called variance error inherent in the model's design. On the other hand, underfit models also lead to poor predictive power, but in their case, this is due to the model being too simplistic and not being able to abstract enough during the training phase. This is due to what's called bias error inherent in the model's design. Generally, as the model's complexity increases, the errors associated with variance increase, but the errors associated with bias decrease. The bias-variance tradeoff implies that there should be a sweet spot at which point more complexity will result in overfitting and less complexity will result in underfitting.[15]

Bonferroni's Principle

There are limits and upper bounds in relation to the accuracy of all models. Generally as the dataset used in modeling grows beyond optimal limits in terms of variables and records, there's a greater chance that the patterns discovered in the data are actually meaningless. As Leskovec, Rajaraman and Ullman note, "These occurrences [patterns] are 'bogus,' in the sense that they have no cause other than that random data will always have some number of unusual features that look significant but aren't.".[16] This is informally called Bonferroni's Principle. In such cases, a model's predicted scores will return more records of significance than would normally be expected, resulting in many false positives.

---

[13] (James, et al. 2013, 25)
[14] (Hawkins 2003, 1)
[15] (Ghatak 2017, 66)
[16] (Leskovec, Rajaraman and Ullman 2014, 5)

Works Cited

Ghatak, Abhijit. *Machine Learning with R.* Singapore: Springer, 2017.

Hawkins, Douglas M. "The Problem of Overfitting." *Journal of Chemical Information and Computer Sciences* 44, no. 1 (2003): 1-12.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning.* New York, NY: Springer, 2013.

Lantz, Brett. *Machine Learning with R.* Birmingham, UK: Packt Publishing Ltd., 2015.

Leskovec, Jure, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets.* 2014.

Reinstein, Ilan. *Random Forests(r), Explained.* n.d. https://www.kdnuggets.com/2017/10/random-forests-explained.html (accessed January 9, 2018).

Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* New York, NY: Cambridge University Press., 2014.

Williams, Graham. *Data Mining with Rattle and R.* New York, NY: Springer, 2011.